2023 Project (no more than 110 points)

This project requires the development of a smartwatch program using the STM32 development board. The program should include a chat component, a calculator component, a photo album component, and an initial interface for users to choose which component to use.

1 Basic Framework (5)

In this section, you need to implement a basic framework that includes an initial interface for users to choose which component to use, and users should be able to return to the initial interface from the components.

1.1 Initial Interface (3)

You need to implement an initial interface (using a single color for simplicity).

The initial interface should include basic configuration information such as displaying the current time and user information (displaying pre-set information). Additionally, users should be able to enter the interface of the component they want to use using the touchscreen or other reasonable means.

1.2 Return to Initial Interface (2)

You need to ensure that users can return to the initial interface after using the component, through some reasonable interaction method (except for using the reset button), without affecting the user's ability to reuse the components.

2 Chat Component (45)

In this section, you need to utilize the development board's 2.4G communication module to achieve data exchange between two development boards and implement the chat functionality.

In the chat component, we need to differentiate between user's **online** and **offline** statuses. In this project, we define the user's online and offline statuses as follows: when a development board is powered on and enters the smartwatch interface, it is considered online, even if the user has not entered the chat component. When a development board exits the main interface of the smartwatch program, it is considered offline.

2.1 Chat Input (2)

You need to implement text input using the serial communication feature. Specifically, users can input chat content through XCOM or other serial communication tools.

2.2 Chat Information Transfer (10)

2.2.1 Message Transfer (5)

You need to transfer received chat content to the target development board using the 2.4G communication module of the development board. The chat information needs to be transmitted between the development boards in a peer-to-peer manner.

2.2.2 Status Detection (5)

Due to the limitations of the development board's 2.4G communication module, you need to detect the status of the other party to ensure the normal transmission of messages.

If the other party is offline, there should be an appropriate prompt indicating that the other party is offline.

2.3 Chat Interface (8)

In this part, you need to use the development board's LCD display to display the chat content.

2.3.1 Basic Display Functionality (3)

Basic display functionality requires you to display the conversation information between the user and other users in the chat component.

When the other party is online and has successfully entered the chat component, you need to display the conversation content along with the sender of each message. The sender information can be the username or a distinctive identifier such as an avatar. The username can be obtained from the initial interface.

2.3.2 Line Break Display (2)

When the content sent by the user is too long to fit on a single line of the development board's screen, you need to have a reasonable interface and line break rules to allow the content to be displayed correctly on multiple lines.

2.3.3 Page Navigation Functionality (3)

When there is a large amount of chat content between users, it may exceed the display capacity of the development board's screen. You need to have a reasonable page navigation functionality (such as left-right scrolling or up-down sliding) to allow users to view all the chat information during their current online session.

2.4 Chat Invitation (8)

In this part, we want you to further improve the chat component so that users can also pay attention to the status of the chat objects while using other components.

2.4.1 Chat Object Information Storage (2)

During the user's current session of running the smartwatch program until exiting the program, you need to store all user information who have had a chat history with the current user, to facilitate the implementation of subsequent functionalities.

2.4.2 User Offline Notification (2)

In this part, when the user finds that a chat object has gone offline, you need to display a notification indicating that the other party is offline and prevent the current user from entering the chat interface with the offline user.

2.4.3 User Online Notification (2)

In this part, we want to implement a feature that notifies the local user when a chat object transitions from offline to online status. This allows the local user to be aware of the chat object's status while using other components.

The trigger condition for the online notification is when the previous chat object starts running the smartwatch program.

2.4.4 Inviting Users to Chat (2)

When a user discovers that a chat object is online, they can initiate a chat invitation. This allows the invited user to receive the chat invitation even while using other components, reminding them to participate in the chat.

2.5 Group Chat (8)

In this section, we would like you to implement the functionality for group chats on three development boards. Note that this feature requires at least three development boards and three communication modules. Having more than three development boards will not earn you additional points.

2.5.1 Individual Chats in a Group (4)

When there are two or more users simultaneously online among the chat objects stored locally, users can choose and specify other users for one-on-one chats.

This functionality requires the interface to display at least one user list and independent chat interfaces related to the selected chat objects.

2.5.2 Group Chats (4)

When there are two or more users simultaneously online among the chat objects stored locally, users can engage in multi-person chats similar to group chats.

For simplicity, you can assume that there can only be one group chat among the three users, and it is not possible to create multiple two-person group chats. In other words, if your program supports group chats, you can provide a button to enter a chat interface that includes all online users.

2.6 Emoji (4)

In this section, we want users to be able to send emojis during chats to enhance the richness of the conversation.

Note that emojis here refer to users selecting emojis from a predetermined list, rather than inputting emojis in real-time in any format.

2.7 Full-Color Interface (5)

All interfaces within this component need to include color elements to receive full points for this section. To reduce code complexity, you can set only certain components as full-color within a single page.
Note: Emoji expressions (if implemented) must be displayed in full color.

3 Calculator Component (25)

In this section, you will utilize the development board's resistive touch screen and LCD screen display functions to create a simple decimal calculator.

3.1 Interaction (1)

In the calculator component, users can only input information to the development board via the resistive touch screen.

3.2 Calculation Functions (24)

3.2.1 Basic Calculation Functions and Display (5)

This functionality requires the ability to perform basic addition, subtraction, multiplication, and division operations, with decimal results displayed.

The calculator should include at least the following buttons for users to perform basic operations: 0, 1, 2, 3, =, +, -, *, /, C (clear). The calculator needs to handle basic error detection, such as division by zero.

The user's input should be displayed at the top of the screen. For example, if the user wants to calculate "3+2", the screen should display "3+2". In case of calculation errors or input errors, "Error" should be displayed. Due to screen size limitations, the user's input formula cannot be infinitely long. You can impose appropriate restrictions based on your own page layout, but the user should be able to input at least 10 characters.

When the user presses the "=" button, the calculation result should be displayed.

There can be multiple ways to display the calculation result. For example, you can send it as text via serial communication to a computer or display it on the LCD screen. Any form of display will not earn you additional points or deductions.

3.2.2 Parentheses, Exponents, and Display (5)

This functionality requires support for parentheses to change the order of operations and the ability to input arbitrary exponents.

When exponent functionality is supported, 2^2 should be represented as "2^2" in the calculator component, representing 2 squared.

When the "=" button is pressed, the calculation result should be displayed.

Note: To maintain consistency, please use the "^" symbol as the button label for the exponent functionality.

3.2.3 Equation Solving and Display (5)

This functionality requires the ability to solve quadratic or linear equations.

When equation solving is supported, the user should input equations in the form of "x-2=0", and the display should show "x=2" as the result, where x is the unknown variable of the equation.

When the "Confirm" button is pressed, the calculation result should be displayed.

For solving the equation y^2=4, the input should be displayed as "y^2 = 4", and the result should be displayed as "y = 2 | y = -2".

Note: This functionality can be combined with the basic calculations in the same interface or implemented in a separate interface. Either approach will not earn you additional points or deductions.

3.2.4 Binary Operations and Display (5)

This functionality requires support for binary operations and provides a separate interface for users to perform binary calculations. The independent interface should only include the buttons 0, 1, =, +, -, *, C, and should not include the functionalities from 3.2.2 and 3.2.3.

When the "=" button is pressed, the calculation result should be displayed.

3.2.5 Long Display (4)

When the user's input formula exceeds the page limit, the user should be able to continue entering. Additionally, to view the entire input formula, the user should be able to scroll the displayed formula left or right by dragging the screen or using other reasonable methods.

4 Photo Album (15)

In this section, you will use the development board's SD card reading component and an SD card to implement the functionality to read and display pictures stored in the SD card, creating a photo album.

4.1 Reading and Displaying Photos (10)

You need to be able to read JPEG format pictures from the SD card and display them in full color. Additionally, you should provide the ability to switch between photos, allowing users to swipe the screen or use serial communication to perform picture switching operations.
Hint: The pictures are stored in the SD card under "/PICTURE".

4.2 Smart Display Functionality (5)

4.2.1 Displaying the Total Number of Photos (2)

In the photo album interface, users should be able to see the total number of photos in the album.

4.2.2 Infrared Remote Control for Photo Switching (3)

Use an infrared remote control to implement the functionality to switch and display photos.

5 Additional Features (no more than 20)

In this section, you can add additional components that you believe are reasonable or add interesting functionalities to the existing components. For example, you can add a game component to implement a simple touch-screen game. Or you can add functionalities to the existing components. The score for this section will be determined through discussions among the evaluators, and the total score for this section will not exceed 20 points.

6 Report (10)

In this section, you need to complete the sections in the document "CS301-2023FALL-lab-project-report-GROUPNUM.doc".

Note: Only one report needs to be completed and submitted per group, and the score for this section will be shared among group members based on their contributions.